

CLAIMS

What is claimed is:

1. A method for instrumenting a hardware description language (HDL) design entity that is generated utilizing a HDL source code file within the syntax convention of a platform HDL, said method comprising:

6 describing an instrumentation entity within said HDL
7 source code file utilizing a non-conventional syntax
8 comment such that said instrumentation entity is embedded
9 within said design entity without being incorporated into
10 an overall design in which said design entity is
11 incorporated.

12 2. The method of claim 1, wherein said HDL source code
13 file includes a description of at least one operating
14 event within the conventional syntax of said platform
15 HDL, and wherein said describing step further comprises
16 associating said instrumentation entity with said at
17 least one operating event utilizing a non-conventional
18 syntax comment within said HDL source code file.

1 3. The method of claim 2, wherein said conventional
2 syntax description of said at least one operating event
3 includes a description of at least one design entity
4 element, said associating step further comprising
5 specifying said design entity utilizing a non-
6 conventional syntax comment.

1 4. The method of claim 1, wherein said describing step
2 further comprises declaring a name associated with said
3 instrumentation entity utilizing a non-conventional
4 syntax comment within said HDL source code file.

1 5. The method of claim 1, wherein said describing step
2 further comprises declaring the presence of an embedded
3 instrument within said HDL source code file utilizing a
4 non-conventional syntax comment within said HDL source
5 code file.

1 6. The method of claim 5, further comprising compiling
2 said HDL source code file using a compiler that
3 interprets said non-conventional syntax comment such that
4 said compiler recognizes said declared presence of said
5 embedded instrument.

1 7. The method of claim 6, further comprising detecting
2 one or more non-conventional syntax comments indicating
3 the presence of one or more embedded instruments during
4 said compiling step.

1 8. The method of claim 7, wherein during a model build
2 process, said compiler translates said design entity HDL
3 source code file into a compiled proto data structure,
4 said method further comprising setting an embedded
5 instrument flag within said compiled proto data structure
6 in response to detecting a non-conventional syntax
7 comment indicating the presence of an embedded
8 instrument.

1 9. The method of claim 8, wherein upon compilation of
2 said design entity HDL source code file into said
3 compiled proto data structure, said proto data structure
4 is passed to an instrumentation load tool, whereupon said
5 method further comprises determining whether said
6 embedded instrument flag is set within said proto data
7 structure.

1 10. The method of claim 9, wherein in response to
2 determining that said embedded instrument flag is set,
3 said method further comprises opening and parsing said
4 design entity HDL source code file to locate one or more
5 embedded instrumentation entity comments.

1 11. The method of claim 10, further comprising
2 generating a proto data structure and an instance data
3 structure corresponding to said instrumentation entity in
4 accordance with said embedded instrumentation entity
5 comments.

1 12. The method of claim 9, wherein upon passing said
2 proto data structure to said instrumentation tool, said
3 method further comprises removing said embedded
4 instrument flag from said proto data structure.

1 13. A program product for instrumenting a hardware
2 description language (HDL) design entity that is
3 generated utilizing a HDL source code file within the
4 syntax convention of a platform HDL, said program product
5 comprising:

6 instruction means for describing an instrumentation
7 entity within said HDL source code file utilizing a non-
8 conventional syntax comment; and

9 instruction means for translating said non-
10 conventional syntax comment such that said
11 instrumentation entity is embedded within said design
12 entity without being incorporated into an overall design
13 in which said design entity is incorporated.

1 14. The program product of claim 13, wherein said HDL
2 source code file includes a description of at least one
3 operating event within the conventional syntax of said
4 platform HDL, and wherein said program product further
5 comprises instruction means for associating said
6 instrumentation entity with said at least one operating
7 event utilizing a non-conventional syntax comment within
8 said HDL source code file.

1 15. The program product of claim 14, wherein said
2 conventional syntax description of said at least one
3 operating event includes a description of at least one
4 design entity element, said program product further
5 comprising instruction means for specifying said design
6 entity utilizing a non-conventional syntax comment.

1 16. The program product of claim 13, further comprising
2 instruction means for declaring a name associated with
3 said instrumentation entity utilizing a non-conventional
4 syntax comment within said HDL source code file.

5 17. The program product of claim 13, further comprising
6 instruction means for declaring the presence of an
7 embedded instrument within said HDL source code file
8 utilizing a non-conventional syntax comment within said
9 HDL source code file.

1 18. The program product of claim 17, further comprising
2 instruction means for compiling said HDL source code file
3 using a compiler that interprets said non-conventional
4 syntax comment such that said compiler recognizes said
 declared presence of said embedded instrument.

1 19. The program product of claim 18, further comprising
2 instruction means for detecting one or more non-
3 conventional syntax comments indicating the presence of
4 one or more embedded instruments during said compiling.

1 20. The program product of claim 19, wherein during a
2 model build process, said compiler translates said design
3 entity HDL source code file into a compiled proto data
4 structure, said program product further comprising
5 instruction means for setting an embedded instrument flag
6 within said compiled proto data structure in response to
7 detecting a non-conventional syntax comment indicating
8 the presence of an embedded instrument.

1 21. The program product of claim 20, wherein upon
2 compilation of said design entity HDL source code file
3 into said compiled proto data structure, said proto data
4 structure is passed to an instrumentation load tool,
5 whereupon said program product further comprises
6 instruction means for determining whether said embedded
7 instrument flag is set within said proto data structure.

1 22. The program product of claim 21, wherein in response
2 to determining that said embedded instrument flag is set,
3 said program product further comprises instruction means
4 for opening and parsing said design entity HDL source
5 code file to locate one or more embedded instrumentation
6 entity comments.

1 23. The program product of claim 22, further comprising
2 instruction means for generating a proto data structure
3 and an instance data structure corresponding to said
4 instrumentation entity in accordance with said embedded
5 instrumentation entity comments.

1 24. The program product of claim 21, wherein upon
2 passing said proto data structure to said instrumentation
3 tool, said program product further comprises instruction
4 means for removing said embedded instrument flag from
5 said proto data structure.

25. A system for instrumenting a hardware description language (HDL) design entity that is generated utilizing a HDL source code file within the syntax convention of a platform HDL, said system comprising:

processing means for describing an instrumentation entity within said HDL source code file utilizing a non-conventional syntax comment; and

processing means for translating said non-conventional syntax comment such that said instrumentation entity is embedded within said design entity without being incorporated into an overall design in which said design entity is incorporated.

26. The system of claim 25, wherein said HDL source code file includes a description of at least one operating event within the conventional syntax of said platform HDL, and wherein said system further comprises processing means for associating said instrumentation entity with said at least one operating event utilizing a non-conventional syntax comment within said HDL source code file.

27. The system of claim 26, wherein said conventional syntax description of said at least one operating event includes a description of at least one design entity element, said system further comprising processing means for specifying said design entity utilizing a non-conventional syntax comment.

28. The system of claim 25, wherein said processing

2 means for describing and instrumentation entity further
3 comprises processing means for declaring a name
4 associated with said instrumentation entity utilizing a
5 non-conventional syntax comment within said HDL source
6 code file.

1 29. The system of claim 25, wherein said processing
2 means for describing an instrumentation entity further
3 comprises processing means for declaring the presence of
4 an embedded instrument within said HDL source code file
5 utilizing a non-conventional syntax comment within said
HDL source code file.

30. The system of claim 29, further comprising
a compiler that interprets said non-conventional syntax
comment such that said compiler recognizes said declared
presence of said embedded instrument within said design
entity.

31. The system of claim 30, further comprising
processing means for detecting one or more non-
conventional syntax comments indicating the presence of
one or more embedded instruments during compilation of
said design entity.

32. The system of claim 31, wherein during a model build
process, said compiler translates said design entity HDL
source code file into a compiled proto data structure,
said system further comprising processing means for
setting an embedded instrument flag within said compiled
proto data structure in response to detecting a non-
conventional syntax comment indicating the presence of an

8 embedded instrument.

1 33. The system of claim 32, wherein upon compilation of
2 said design entity HDL source code file into said
3 compiled proto data structure, said proto data structure
4 is passed to an instrumentation load tool, whereupon said
5 system further comprises processing means for determining
6 whether said embedded instrument flag is set within said
7 proto data structure.

1 34. The system of claim 33, wherein in response to
2 determining that said embedded instrument flag is set,
3 said system further comprises processing means for
4 opening and parsing said design entity HDL source code
5 file to locate one or more embedded instrumentation
6 entity comments.

1 35. The system of claim 34, further comprising
2 processing means for generating a proto data structure
3 and an instance data structure corresponding to said
4 instrumentation entity in accordance with said embedded
5 instrumentation entity comments.

1 36. The system of claim 33, wherein upon passing said
2 proto data structure to said instrumentation tool, said
3 system further comprises processing means for removing
4 said embedded instrument flag from said proto data
5 structure.